Agile meets CMMI: Culture clash or common cause?

Richard Turner[1], Apurva Jain[2]

[1]The George Washington University, Department of Engineering Management and Systems Engineering, School of Engineering and Applied Sciences, Washington, DC 20052
turner@seas.gwu.edu
[2] University of Southern California, Center for Software Engineering, Computer Science Department, Los Angeles, CA 90089-0781
apurvaj@sunset.usc.edu

Abstract. This paper is based on a workshop held at the University of Southern California Center for Software Engineering in March, 2002. The components of the Capability Maturity Model Integration℠ (CMMI℠) Systems Engineering/ Software Engineering/Integrated Product and Process Model[3] are evaluated for their support of agile methods. We also present a set of dualistic concepts differentiating the approaches. The results help to identify and reduce the level of mythology and establishes common ground for further discussion.

1. Introduction

If one were to ask a typical software engineer if the Capability Maturity Model® for Software (CMM®) [1] and process improvement were applicable to agile methods, the response would most likely range from a blank stare to hysterical laughter. Although attempts to reconcile the positions appear in the literature [2-4], the two approaches have been informally characterized as having the same relationship as oil and water. In a recent workshop, the idea of insurmountable differences was challenged. The result of the challenge was an exercise that compared each component of the Capability Maturity Model Integration Systems Engineering, Software Engineering and Integrated Process and Product Development (CMMI℠-SE/SW/IPPD) model [5, 6] with agile concepts. Additional characterizations of the two approaches, beyond oil and water, were also developed and a survey of the level of agreement performed.

2. A short primer on process improvement and CMMI

Process improvement grew out of the quality movement and the work of Crosby [7], Deming [8], and Juran [9], and is aimed at increasing the capability of work

[3] The following Carnegie Mellon University service marks and registered marks are used in this paper: Capability Maturity Model®, CMM®, CMM Integration℠, and CMMI℠.

processes. By increasing the capability of its processes, an organization becomes more mature and so operates at a higher level of effectiveness.

One means of achieving this focus on process is by using a capability model to guide and measure the improvement. Assessments against the model provide findings that initiate corrective actions which result in better processes. Models often are organized so that there is a proven, well-defined order by which processes are improved based on the experience of successful projects and organizations. The first model of this type was the Software Engineering Institute's Capability Maturity Model for Software. The latest in capability model thinking is represented in the Capability Maturity Model Integration (CMMI) effort and the product suite it has developed.

CMMI is essentially a set of requirements for engineering processes, particularly those involved in product development. It consists of two kinds of information – process areas (PAs) that describe the goals and activities that make up process requirements in a specific focus area, and generic practices (GPs) that are applied across the process areas to guide improvement in process capability.

The process areas include requirements for

- basic project management and control
- basic engineering life cycle processes
- fundamental support processes
- process monitoring and improvement processes (similar to SW-CMM)
- integrated development using teams

The second type of information CMMI provides is a set of generic practices that support the improvement of the processes established under the Process Areas. The generic practices are associated with a six-level capability scale that describes relative capabilities as follows:

0. Not performed (Not even doing the basics)
1. Performed (just doing it)
2. Managed (fundamental infrastructure to accomplish the process generally at the project level)
3. Defined (institutionalizes a standard version of the process for tailoring by projects)
4. Quantitatively managed process (uses quantitative measures to monitor and control selected sub-processes)
5. Optimizing (constant adaptation of processes based on quantitative measures)

CMMI users apply these two kinds of information to establish, refine and manage the processes used to meet organizational goals.

3. Methodology

A workshop on agile methods was held as part of the annual review of the research conducted by the Center for Software Engineering, located at the University of Southern California. Over 40 participants attended, including researchers, research sponsors, and affiliates as well as invited experts on agile methods. One of four breakout groups was asked to look at agile methods in the context of CMMI and

process improvement. This sub-group included members from government research facilities, developers, a government sponsor, an agile methods expert, and academics. The sub-group members had expertise in agile methods and CMMI.

The initial task the sub-group was to classify each CMMI component as in conflict with, of no consequence to, or supportive of agile methods in general. This is essentially the reverse of the approach taken by Robert Glass [4]. The second task was to identify and capture significant conceptual differences and similarities between the two approaches. This was accomplished by brainstorming, discussion, selection, and revision. The Agile Manifesto [10] served as the basis for analysis, but information from practitioners of a particular agile method was included if it seemed relevant. Realizing that agile methods differ in many ways, this approach was viewed as an expedient way to resolve issues given the time constraints of the break-out group.

The sub-group results were reported to the larger body provoking considerable discussion. In light of this response, an informal survey was developed and distributed to the workshop participants to roughly measure the degree of agreement with the break-out group's findings. The survey, made available a week after the conference, was completed by 19 of the attendees. No demographic information was collected.

4. Component comparison

The CMMI components considered by the sub-group were Process Areas and Generic Practices. For each component, a finding was determined, characterized as follows:

- *Conflicts (C)*. The CMMI requirement is a barrier to implementing agile methods
- *Neutral (N).* The CMMI requirement does not impact implementing agile methods
- *Supports (S)*. The CMMI requirement is an enabler to implementing agile methods

Where the survey results showed no majority, the two findings with the largest percentage are indicated separated by a dash (e.g. C-N for Conflicts and Neutral)

The degree of agreement is based on the results of the informal survey and is scaled according to the percentage of survey respondents that agreed with the finding:

- *Strong (S).* 75% or greater
- *Medium (M).* 50% to 74%
- *Low (L)*. 25% to 50%
- *None (N).* Below 25%

Tables 1 and 2 summarize the results of the Agile methods to CMMI component mapping. Notes on the individual component findings may be found in the appendix.

Table 1. Agile method vs. CMMI Process Area conflict findings

Process Area	Sub-Group Finding	Survey Finding	Agreement
Organizational Process Focus	C	C	M
Organizational Process Definition	S	C-N	N
Organizational Training	S	N-S	L
Organizational Process Performance	C	C	M
Organizational Innovation and Deployment	C	C-S	L
Project Planning	S	S	M

Process Area	Sub-Group Finding	Survey Finding	Agreement
Project Monitoring and Control	N	S	L
Supplier Agreement Management	N	N	M
Integrated Project Management	S	S	M
Risk Management	S	N	L
Integrated Teaming	S	S	H
Quantitative Project Management	N	C	N
Requirements Management	S	N	L
Requirements Development	S	S	M
Technical Solution	S	S	M
Product Integration	S	S	M
Verification	S	S	M
Validation	S	S	M
Configuration Management	N	None	L
Process and Product Quality Assurance	N	C-N	L
Measurement and Analysis	N	C-N	L
Decision Analysis and Resolution	C	C	M
Organizational Environment for Integration	S	S	M
Causal Analysis and Resolution	N	N	M

Table 2. Agile method vs. CMMI Generic Practice conflict findings

CMMI Generic Practices	Sub-Group Finding	Survey Finding	Agreement
2.1 Establish an Organizational Policy	N	N-S	L
2.2 Plan the Process	S	N-S	L
2.3 Provide Resources	S	N-S	L
2.4 Assign Responsibility	S	S	M
2.5 Train People	S	N	L
2.6 Manage Configurations	S	C-S	L
2.7 Identify and Involve Relevant Stakeholders	S	S	H
2.8 Monitor and Control the Process	N	N	M
2.9 Objectively Evaluate Adherence	C	C	M
2.10 Review Status with Higher Level Management	N	N-S	L
3.1 Establish a Defined Process	N	C	N
3.2 Collect Improvement Information	C	C	M
4.1 Establish Quantitative Objectives for the Process	C	N	L
4.2 Stabilize Subprocess Performance	C	C-N	L
5.1 Ensure Continuous Process Improvement	S	C-N	L
5.2 Correct Root Causes of Problems	N	N	M

4.1 Summary of CMMI Component Comparison

Of the 40 components analyzed by the sub-group and validated by the larger group, the results can be summarized as follows:

- 7 components are seen as clearly in conflict
- 10 components are seen as possibly in conflict
- 11 components are seen as clearly supportive
- 11 components are seen as no worse than neutral

- 1 component had no consensus finding

Only 17 of the 40 components are considered in conflict or possible conflict. Twenty two components were seen to be supportive of or neutral to agile methods. The components that were deemed in conflict were primarily those that addressed organizational process. This makes sense given the project focus of agile methods. Many of the supportive components were fundamental project management activities which must be performed in some fashion for any successful project. Agile activities also mapped well to the product development activities. The generic practices were mixed in support, which we believe reflects their process improvement focus.

4.2 Inter-group dissention

It should be noted that the component mappings represents multiple perceptions from both sides of the divide. There were two distinct groups from the CMMI school – a conservative, by-the-letter group and a liberal, concepts-oriented group. Likewise, there were "conservative" agilists who were extremely rigid in their definitions and liberal agilists who saw the value of comparisons and hybridization. This is born out in the outlier data on both sides that prevented full consensus to be reached.

4.3 CMMI component interaction

The results ignore the interactions between the GPs and the PAs, such as enabling PAs like CM, that provide ways to implement the GPs (CM is a good example). It might have been better to have paired these in order to show their association. In a similar vein, the IPPD extension impacts nearly all of the PAs and how they are accomplished. If we had explicitly modified the purpose and goal language to express the IPPD implications of integrated development and team relationships, the results might have shown more supportive or neutral components.

5. Conceptual Comparison

The sub-group defined a number of conceptual characterizations of the differences and similarities of agile methods and CMMI. Some of these were obvious, but others indicated fundamental differences that are not as immediately evident. The survey of the larger group asked the respondent to identify the level to which they agreed with each of the characterizations on a scale from 1 to 7, with 1 representing total disagreement.

5.1 Differences

In looking at the two paradigms, the sub-group identified a number of areas where CMMI and agile had strikingly different and often quite enlightening characterizations. Survey results show that there is general (although not always strong) agreement from the respondents with the workgroup's characterizations. The following paragraphs discuss each of the differences.

What provides customer trust:

CMMI - Process Infrastructure
AGILE - Working S/W, Participation
Agreement level: 5.7 out of 7 (.81)

This pair stems from the observation that process people count on their process maturity to provide confidence in their work. CMM appraisals are often used in source selection for large system implementation or for sourcing decisions.

Agile people use the idea of working software and customer participation to instill trust. In proposals they use their track record, the systems they've developed and the expertise of their people to demonstrate capability.

Scope of approach:

CMMI - Broad, Inclusive and Organizational
AGILE - Small, Focused
Agreement level: 5.6 out of 7 (.80)

This observation is based on the premise that CMMI covers a broader spectrum of activities than agile methods. CMMI looks to develop the capabilities of an organization in a number of disciplines, including systems engineering.

Agile methods are generally used on smaller projects, and concentrate on delivering a software product on time that satisfies a customer.

Where knowledge created during projects (lessons learned, etc.) resides:

CMMI - Process Assets
AGILE - People
Agreement level: 5.8 out of 7 (.83)

The management and retention of knowledge is key to organizational survival. In process-oriented organizations this knowledge is maintained in the process assets (including process definitions, training materials, etc.). This is to support uniformity across projects and comparability for measurement.

Agile methods generally focus on a project rather than an organization and maintain their experience in the people doing the work. As these people work on more and more tasks, that knowledge is shared across the organization.

Practitioner and advocate characteristics:

CMMI - Disciplined, Follow Rules and Risk Averse
AGILE - Informal, Creative and Risk Takers
Agreement level: 5.2 out of 7 (.74)

Here the differences are on the perceived mind set of the practitioners and supporters of the two approaches. CMMI supporters are often characterized as rigid, structured, and bureaucratic. Agile supporters are seen as freer spirits who let their talents flow unfettered into the work and don't worry about exactly how the work gets accomplished, so long as it meets the customer's needs.

Scaling Challenges:

CMMI - Scaling down -- Doable, but Difficult
AGILE - Scaling up -- Undefined

Agreement level: 5.6 out of 7 (.80)

While there can be debate on this issue, CMMI is generally seen as appropriate for large projects and complex organizations, but difficult to apply to small companies or individual teams. Agile is seen as wonderful for small organizations and small, separable projects, but its ability to scale up to larger projects and organizations has been widely questioned.

Operational Organization:

CMMI - Committees
AGILE - Individuals
Agreement level: 4.2 out of 7 (.60)

This observation saw differences in how the two approaches accomplished work. In process-based organizations decisions and work are usually done in a committee, and decision authority and responsibility may be dispersed. Process-based organizations often have highly specific chains of command, decision making processes and requirements, and other operational structures that require multiple sign-offs and long coordination times.

Agile methods are observed to be dependent on the individual to accomplish tasks and the team to make quick, informed, product-oriented decisions. The authority usually resides in the team doing the work, and there is little bureaucracy.

It should be noted that this had the lowest agreement level from the survey. The reason may be in the summary words used in the survey which could be considered to negate the collaborative nature of agile methods.

Goals of the approach:

CMMI - Predictability, Stability
AGILE - Performance, Speed
Agreement level: 5.6 out of 7 (.80)

Process maturity (or capability) is focused on predictability and stability. They are aimed at developing the capabilities of organizations rather than specifically delivering products, and seek to enable predictable performance regardless of the staff involved.

Agile methods are focused on speed, performance, and delivering quickly to the customer. Agile practices are more opportunistic in nature – that is, they support rapid change to accommodate situational needs and environmental demands.

Communication style

CMMI - Macro, Organizational
AGILE - Micro, Person to Person
Agreement level: 5.5 out of 7 (.79)

In process-oriented organizations, common processes and training are seen to support communication across broad sections of the organization using work products as the medium. Such artifacts provide traceability, can be used in process analysis, and provide history for later projects. Artifacts also are generally developed in conformance with a standard. This adds additional time and effort to the project.

Agile methods tend to encourage frequent, person-to-person communication and specifically address only intra-project communication. Much of the communication is "as necessary" and has no lasting artifact. This allows rapid development, but can make recovery or later analysis more costly.

Usual focus of issue resolution:

CMMI - Words
AGILE - Product
Agreement level: 5.2 out of 7 (.74)

This was probably the most interesting difference stated. When process people work a problem, there is an enormous amount of energy expended on defining the specifics and finding just the right words for both the problem and a solution. The waterfall approach is evident in the way they consider getting just the right description so that there is agreement and the results can be communicated clearly to a large group.

Agile people tend to act first and talk later so they can get the product out. Rather than discuss an issue at length, people generally try something and see if it works. If it doesn't, keep trying until something does. The spiral or evolutionary nature of their thinking leads to a number of trial solutions which may refine the understanding of the problem.

5.2 Similarities

The sub-group identified two places where the CMMI and agile methods found common ground.

Both have specific rules

Agreement level: 4.6 out of 7 (.66)

It became obvious early on in the discussions that process-oriented and some of the agile methods (particularly XP) have specific rules that must be followed. Agile tended to have considerably fewer rules, but the feeling of the group was that this made those few much more critical.

Both are motivated by the desire to become a high performance organization

Agreement level: 4.4 out of 7 (.63)

Both approaches are motivated to develop and maintain high performance organizations. Process approaches work in a more traditional industrial fashion, using the concepts of engineering and manufacturing to establish a well-defined machine of an organization.

Agile uses more post-industrial ideas. Work is done within the specific context of the problem, and the goal is to establish experts with generalized talent that can form a team and deliver an acceptable product to the customer as quickly as possible.

6. Analysis and conclusions

From the results of the component comparisons it is evident that while there are significant differences, the "oil and water" description of CMMI and agile approaches is somewhat overstated. It is also clear that while both represent methodologies of a sort – maturity models and appraisal techniques or agile tenets and practices – the defining characteristic is the attitude or mindset under which development activities are accomplished.

All of the conceptual comparisons were validated by the survey. While there were still outliers, our general opinion is that the differences between the agile and process worlds are beginning to be better articulated and so better understood. Because this data is based primarily on perceptions, an empirical analysis of the conceptual pairs to determine the validity of the findings is currently under way by the authors.

It is our belief that there is much in common between the two world views, and that the strengths and weaknesses are often complimentary. We also believe that neither way is the "right" way to develop software or software-intensive systems. Rather, there are instances of projects or phases of projects when one or the other represents a significant advantage. While development organizations will almost certainly have a preferred manner of doing business, they should be able to identify and respond to these instances by adapting their work processes to the work at hand. We look forward to continuing discussion and the results of collaborative efforts and hybrid methods that are sure to appear in the near future.

7. References

1. Paulk, M.C.e.a., *Capability Maturity Model for Software, Version 1.1*. 1993, Software Engineering Institute, Carnegie Mellon University: Pittsburgh, PA.
2. Paulk, M., *Extreme Programming from a CMM Perspective.* IEEE Software, 2001.
3. Boehm, B.W., *Get Ready for Agile Methods, with Care.* IEEE Computer, 2002.
4. Glass, R.L., *Agile versus Traditional: Make Love not War*. Cutter IT Journal, 2001. **Vol. 14, No. 12**(December): p. 12-18.
5. Ahern, D.M., A. Clouse, and R. Turner, *CMMI distilled : a practical introduction to integrated process improvement*. The SEI series in software engineering. 2001, Boston: Addison-Wesley. xv, 306.
6. CMMI Development Team, *CMMI-SE/SW/IPPD, V1.1 : Capability Maturity Model Integrated for Systems Engineering, Software Engineering and Integrated Product and Process Development, Version 1.1 : continuous representation.*. 2001, Software Engineering Institute, Carnegie Mellon University: Pittsburgh, PA. p. 688.
7. Crosby, P.B., *Quality is Free: The Art of Making Quality Certain*. 1979, New York, NY: MacGraw-Hill.
8. Deming, W.E., *Out of Crisis*. 1986, Cambridge, MA: MIT Center for Advanced Engineering.

9. Juran, J.M., *Juran on Planning for Quality*. 1988, New York, NY: MacMillan.
10. Agile Alliance, *Manifesto for Agile Software Development*. 2001.
11. Schwaber, K. and M. Beedle, *Agile Software Development with Scrum*. 2002, Upper Saddle River, N. J.: Prentice Hall.

8. APPENDIX – Notes on CMMI Component Findings

CMMI Process Areas

Organizational Process Focus (Findings: C, C Agreement: M)

Conflict was based on the implied infrastructure needed to accomplish the goals. While agile processes evolve, they do so under their own experience, generally within a team or a project. Agile organizations do generally have an organizational process (the agile method or ecosystem) within which improvements take place.

Organizational Process Definition (Findings: S, C-N Agreement: N)

This process area caused some difficulty across the broader group. The sub-group accepted the idea of an agile method or ecosystem being essentially a process asset repository, albeit perhaps an informal one. While the larger group did not agree, it was evenly divided between conflict and neutrality.

Organizational Training (Findings: S, N-S Agreement: L)

Agile methods rely on practitioners trained in the method. The agile manifesto values individuals over processes and tools, and so should lead toward training and mentoring. There is no requirement in the PA for complex infrastructure – simply that the capability exists and is maintained.

Organizational Process Performance (Findings: C, C Agreement: M)

The idea of measuring a process and maintaining baselines and models was certainly in conflict with the agile manifesto. A full 25% of the larger group indicated CMMI supported agile in this area. One comment during the brief out indicated that some of the agile methodologies, Scrum [11] for example, have metrics which could be characterized as process metrics.

Organizational Innovation and Deployment (Findings: C, C-S Agreement: L)

There was considerable discussion about this PA. Several argued that it captured the essence of agile development. Other's cited the need for infrastructure and the organizational focus implied in the process area goals. There was nearly a 50/50 split between conflicting and supporting among the wider group.

Project Planning (Findings: S, S Agreement: M)

Most agile methods require a high level of start-up planning and risk assessment. The sub-group indicated this as one of the similarities between the two development approaches.

Project Monitoring and Control (Findings: N, S Agreement: L)

The difference of opinion results from the level of tracking and planning. Some thought the PA implied more rigorous planning and tracking than agile methods usually employ.

Supplier Agreement Management (Findings: N, N Agreement: M)

Almost everyone saw this as something rarely addressed in agile projects due to the nature of the teams and the development focus of the work.

Integrated Project Management (Findings: S, S Agreement: M)

Agile methods are generally team-based and integrate the developers, validators, and customers. Over 90% of the respondents indicated neutral or supportive.

Risk Management (Findings: S, N Agreement: L)

Most agile methods are designed to mitigate certain types of risks – particularly those from changing requirements and schedules. The opposing view held that agile didn't really address long-term risk and didn't strictly follow the CMMI process of identifying, analyzing and tracking.

Integrated Teaming (Findings: S, S Agreement: H)

Integrated teaming is a key facet of all of the agile methodologies.

Quantitative Project Management (Findings: N, C Agreement: N)

Because agile methods don't necessarily perform these activities, there is nothing in the agile manifesto that precludes their performance. The larger group indicated a conflict, with statistical control, deemed non-agile, was the primary concern.

Requirements Management (Findings: S, N Agreement: L)

Since agile tenets call for continuous interaction with the customers, it was inferred that requirements were being closely managed. The larger group reflected that tracking and plan management did not support agility.

Requirements Development (Findings: S, S Agreement: M)

This PA supports the agile concepts of close customer relationships, customer-based requirements elicitation and stakeholder involvement.

Technical Solution (Findings: S, S Agreement: M)

The only arguments against were based on the requirement for support documentation – something that some agile methodologies don't strictly support.

Product Integration (Findings: S, S Agreement: M)

There was some support for a neutral finding based on the idea that this PA addresses integrating components from hardware and software sources and that agile rarely dealt with that type of project.

Verification (Findings: S, S Agreement: M)

Peer reviews are closely aligned with pair programming. The concept of requirements traceability caused some concern, in that many agile methods focus on functional requirements and place little or no value on nonfunctional requirements or capturing derived requirements which may be lost in later versions.

Validation (Findings: S, S Agreement: M)

The close relationship with the customer in most agile methods is strongly supported by this PA. Concerns were similar to those for the Verification PA.

Configuration Management (Findings: N, None Agreement: L)

The sub-group found configuration management to be neutral, with no consensus in the larger group. Some saw frequent builds as strong configuration management, while others pointed out that CM in CMMI was to be applied as appropriate to all work products, which does not support the agile reduced emphasis on documentation.

Process and Product Quality Assurance (Findings: N, C-N Agreement: L)

Again, the sub-group saw this as neutral while the larger group leaned toward conflict. Given the emphasis on process, non-compliance, and work products, it seemed to some that this PA was focused on peripheral materials rather than product.

Measurement and Analysis (Findings: N, C-N Agreement: L)

The sub-group, pointing out that several agile methods included some form of progress measure, found this PA to be neutral toward agility. Others felt the concept of measurement and analysis was not a part of the agile approach, and meeting the schedule with an acceptable product to the user was sufficient for agilists.

Decision Analysis and Resolution (Findings: C, C Agreement: M)

This PA's focus on establishing specific processes for team functions was in conflict with the spirit of agility. To be agile means to be able to adapt quickly to the situation rather than be bound to pre-conceived criteria and a strict alternative evaluation or decision analysis process.

Organizational Environment for Integration (Findings: S, S Agreement: M)

Most of the agile methods are supported by a person-friendly, "whatever the developer needs" environment which mirrors the CMMI goals. Some were concerned that infrastructure of any type negated agility.

Causal Analysis and Resolution (Findings: N, N Agreement: M)

Agile methods use reflection, refactoring, or other cognitive reviews of the product and process to establish lessons learned and make the next cycle more efficient.

CMMI Generic Practices

GP 2.1 Establish an Organizational Policy (Findings: N, N-S Agreement: L)

While not an important part of agile methods, policy was not necessarily in conflict with their intent. The move to agile often requires support by management that may be enhanced through policy.

GP 2.2 Plan the Process (Findings: S, N-S Agreement: L)

The sub-group generally felt that agile method's up-front activities accomplished this in most cases. The larger group was a bit more skeptical, seeing maintaining a plan for the process (not the project) was not necessary to agility.

GP 2.3 Provide Resources (Findings: S, N-S Agreement: L)

Providing the resources necessary to complete the work is not in conflict with agile values. Some felt it was not a part of the development activity and thus neutral.

GP 2.4 Assign Responsibility (Findings: S, S Agreement: M)

Assignment of responsibilities with the associated authority was strongly supportive of the people performing the work and so supported the agile approach.

GP 2.5 Train People (Findings: S, N Agreement: L)

The sub-group found it supported agility by providing incentives to train developers in the methodology and to mentor new personnel.

GP 2.6 Manage Configurations (Findings: S, C-S Agreement: L)

Like the Configuration Management PA, the practice that requires CM to be applied across processes found conflicting opinions across the participants.

GP 2.7 Identify and Involve Relevant Stakeholders (Findings: S, S Agreement: H)

There was almost no disagreement with this finding.

GP 2.8 Monitor and Control the Process (Findings: N, N Agreement: M)

Within and between cycles, agile methods are monitored for adherence to functional and schedule requirements as established in the plans.

GP 2.9 Objectively Evaluate Adherence (Findings: C, C Agreement: M)

The idea of a process mafia that checked on how the developers developed was seen as significant barrier to agile methods. The sub-group noted, however, that in XP and other more strictly defined methods, there was a sense that the team lead/coach/facilitator performed this function on a person-by-person basis.

GP 2.10 Review Status with Higher Level Management (Findings: N, N-S Agreement: L)

Some expressed the opinion that agile adoption requires executive support, so success story briefings are helpful. Where multiple agile teams provide parts of a product, briefing higher management was seen as a necessity.

GP 3.1 Establish a Defined Process (Findings: N, C Agreement: N)

Some agile methods are detailed while others are more akin to philosophies. Some considered any attempt to define the process as in conflict with agile approaches.

GP 3.2 Collect Improvement Information (Findings: C, C Agreement: M)

The thrust of this practice is to collect information to improve the process.

GP 4.1 Establish Quantitative Objectives for the Process (Findings: C, N Agreement: L)

Quantitative objectives for the process seems not in the spirit of agile concepts.

GP 4.2 Stabilize Subprocess Performance (Findings: C, C-N Agreement: L)

This practice is closely related to statistical process control. Other negative considerations were the concept of sub-process and the establishment of objectives that were not necessarily associated with the customer.

GP 5.1 Ensure Continuous Process Improvement (Findings: S, C-N Agreement: L)

Some saw continuous improvement as a goal of agile methods.

GP 5.2 Correct Root Causes of Problems (Findings: N, N Agreement: M)

The general consensus for this practice was one of neutrality, considering that root cause analysis, while a worthwhile endeavor, was neither recommended nor proscribed by agile methods.